Web-Agile Facial Emotion Recognition and Eye-Tracking System (WAFER-ET)

Mustafa Can Gursesli[1], Febri Abdullah[2], Ruck Thawonmas[2], Andrea Guazzini[3], Antonio Lanata[1]
[1]Department of Information Engineering, University of Florence, Italy
[2]Graduate School, College of Information Science and Engineering, Ritsumeikan University, Japan
[3]Department of Education, Literatures, Intercultural Studies, Languages and Psychology, University of Florence, Italy
mustafacan.gursesli@unifi.it, gr0397fs@ed.ritsumei.ac.jp, ruck@is.ritsumei.ac.jp, andrea.guazzini@unifi.it, antonio.lanata@unifi.it

***Abstract*—This study presents the Web-Agile Facial Emotion Recognition and Eye-Tracking System (WAFER-ET), which combines AI-based facial expression recognition and eye-tracking models. Eye tracking is implemented using WebGazer.js and facial expression recognition using a custom lightweight CNN-based model (CLCM). Experimental analysis has revealed that the system provides real-time processing and data streaming. The latency of the system between the Experimental Platform and the Real-Time Results Platform was measured—and low latency was achieved. The integration of facial expression recognition and eye tracking in the WAFER-ET system offers the possibility of use in different fields of application.**

***Index Terms*—cloud computing, eye-tracking, emotion recognition, REST, real-time data**

I. Introduction

In recent decade, there has been an emerging use of real-time applications in various domains, including medicine, entertainment, and occupational safety. This growth in real-time applications can be attributed to the focus of many disciplines on processing real-time outputs from AI models. Therefore, there is an increasing demand for systems that can facilitate real-time processing for various models. Furthermore, in parallel with the increase in the number of real-time models, there is a demand for functionalities such as the streaming and storing of these real-time outputs to remote users over the internet. This study is built on the system of Gursesli et al. in 2023 [1] and aims to develop a cloud-based real-time system that integrates AI-based facial emotion recognition and eye-tracking models, enabling **W**eb-**A**gile **F**acial **E**motion **R**ecognition and **E**ye-**T**racking (WAFER-ET) streaming and storage system.

II. Related Works

A. Representational State Transfer

Representational State Transfer (REST) is a widely accepted web-based architecture used in both academia and industry [2]. It was introduced as a framework for the design and development of distributed hypermedia systems. REST facilitates component caching to reduce user-perceived latency, enforces security measures, and enables the encapsulation of legacy systems. It relies on the Hyper Text Transfer Protocol (HTTP) for communication between clients and servers. The structure of REST provides benefits such as modifiability and statelessness, which enhance interoperability. These benefits make REST a valuable approach to efficiently manage real-time data. Moreover, it has been used as an application in various studies in literature [3], [4].

B. Convolutional Neural Network

Convolutional Neural Networks (CNNs) are widely used in various recognition tasks, including Facial Emotion Recognition (FER), due to their effectiveness [5]. However, CNNs typically have complex structures and functions that require extensive computational power. Recently, researchers have focused on the improvement of models with lower complexity but higher performance with approaches such as MobileNets [6]. The main reason for focusing on that kind of study is that complex models e.g. AlexNet [7] have low usability in applications due to their large structures.

Eye tracking is a popular research tool for studying various topics such as learning, psychology, and entertainment. The number of eye-tracking models using web-camera is increasing day by day due to technological advances. These models offer less cost and more usability compared to physical eye-tracking systems.

III. Materials and System

A. WebGazer.Js

WebGazer.js is a JavaScript-based eye-tracking algorithm that enables real-time tracking of eye-gaze positions on web platforms using webcams integrated into laptops and mobile devices [8]. The algorithm comprises two essential components: a pupil detector and a gaze estimator. The pupil detector identifies the location of the eye and pupil by leveraging the webcam feed. Simultaneously, the gaze estimator employs regression analysis to estimate the specific location on the screen where the individual is directing their gaze.

B. Custom Lightweight CNN-based Model (CLCM)

CLCM is a lightweight version of MobileNetV2 developed using a transfer learning approach with a reduced number of parameters (3.5M vs 2.4M). This version offers improved efficiency in terms of both computation (GFlops) and inference time compared to the original MobileNetV2 model. Furthermore, for this particular study, CLCM was trained with the RAF-DB dataset [9].

C. System Design

The system is designed with three components under two main modules. The experimental platform (A) and the real-time results platform (B) are in the front-end module. In the back-end module, there is a Server (C) component. The flow of the system can be summarised as follows: Experiment Platform (A) captures the emotions, eye movements, and positions of the subject through an embedded models, converts them into results, and sends them to the Real-time Results Platform (B).

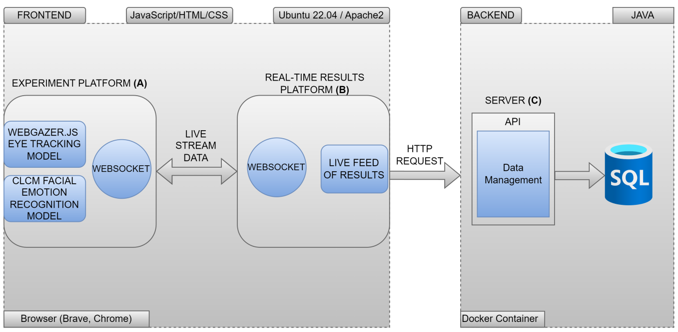

Fig. 1. The Components of the System Design: Experiment Platform (A), Real-Time Results Platform (B), and Server (C).

These two platforms (websites) communicate with each other via WebSockets and provide data flow by constantly listening to exchanged messages from A to B and vice-versa. The emotion recognition and eye-tracking model placed on the experimental platform initiates data collection and its results are then transmitted to a real-time results platform and streamed to the back end via HTTP requests. Following the end of the data collection session, the eye-tracking data streamed instantaneously on the Real-time Results Platform (B), is sent via HTTP request to the Server (C), which constitutes the last stage of the data flow in the back end and storage (see Figure 1).

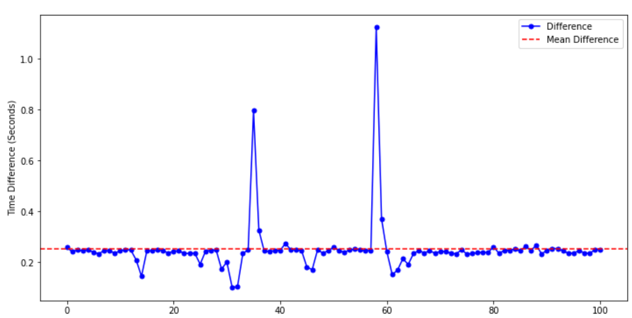

Fig. 2. Delay Time from Platform A to Platform B

IV. Experiment and Result

Experimental session was carried out to assess the reliability of the proposed system architecture in an online scenario. In order to measure the time delay of the online scenario, the timestamps were collected during a 100 seconds time window (see Figure 2). The delay is measured by subtracting the timestamp value of the Experiment Platform (A) from the timestamp received on the Real-Time Results Platform (B), i.e., arrival time - starting time (B-A). Platform (A) sends data to platform (B) at the frequency of 1HZ. According to Descriptive Statistics, the MAD value for the system delay was 0.006, the median value was 0.244, the minimum value was 0.101 and the maximum value was 1.123. Figure 3 shows, the real-life demo of the system.

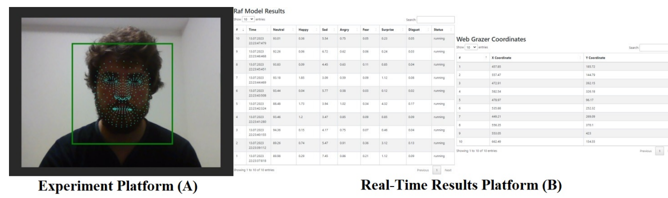

Fig. 3. Real-life demo of the system

V. Conclusion and Future Studies

In this study, we proposed the WAFER-ET system, a cloud-based real-time system that integrates AI-based facial emotion recognition and eye-tracking models. In this implementation, the system uses WebGazer.js for eye-tracking and a custom lightweight CNN-based model (CLCM) for facial emotion recognition. The experimental session evaluated the performance of the proposed system architecture in an online scenario. The recognition of emotions and eye movements is crucial in various scientific disciplines. In particular, real-time processing of these variables has become an important aspect in both academic and industrial research. Future investigations should focus on exploring the use of real-time cloud-based systems, such as the present study, and adapting them to different experimental settings.

References

[1] M. C. Gursesli, M. E. Selek, M. O. Samur, M. Duradoni, K. Park, A. Guazzini, and A. Lanat, “Design of cloud-based real-time eye tracking monitoring and storage system,” *Preprints.org*, June 2023.

[2] R. T. Fielding, *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.

[3] M. Wedel, R. Pieters *et al.*, “Eye tracking for visual marketing,” *Foundations and Trends® in Marketing*, vol. 1, no. 4, pp. 231–320, 2008.

[4] A. Ehsan, M. A. M. Abuhaliqa, C. Catal, and D. Mishra, “Restful api testing methodologies: Rationale, challenges, and solution directions,” *Applied Sciences*, vol. 12, no. 9, p. 4369, 2022.

[5] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, “A convolutional neural network cascade for face detection,” in *Proceedings of the IEEE conference on computer vision*, 2015, pp. 5325–5334.

[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

[8] A. Papoutsaki, J. Laskey, and J. Huang, “Searchgazer: Webcam eye tracking for remote studies of web search,” in *Proceedings of the 2017 conference HIIA*, 2017, pp. 17–26.

[9] S. Li, W. Deng, and J. Du, “Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2852–2861.